	<p style="text-align: center;">Metody programowania 2023/2024</p> <p style="text-align: center;">Maksymalna podtablica 2D</p> <p style="text-align: center;">o najmniejszej liczbie elementów</p>	<p style="text-align: center;">P_01</p>
---	--	---

Opis

Dla danej niepustej tablicy dwuwymiarowej liczb całkowitych: $a[0][0], \dots, a[n-1][m-1]$ dla $0 \leq i \leq j < n$, $0 \leq k \leq l < m$ definiujemy jej maksymalną podtablicę $mstab = a[i..j][k..l]$ jako spójny fragment podtablicy o maksymalnej nieujemnej sumie elementów, obliczanej według wzoru:

$msum(i, j, k, l) = 3*D + 2*U$ gdzie:

D = suma dodatnich elementów $a[x][y]$ tej podtablicy, dla których $i \leq x \leq j$ oraz $k \leq y \leq l$.

U = suma ujemnych elementów $a[x][y]$ tej podtablicy, dla których $i \leq x \leq j$ oraz $k \leq y \leq l$.

W przypadku, gdy wszystkie elementy tablicy są mniejsze od zera, maksymalna podtablica jest pusta i $msum$ jest nie określona. Gdy wszystkie elementy tablicy są równe 0 to maksymalna podtablica składa się z elementu $a[0..0][0..0]$ i $msum(i, j, k, l)$ - jest równa się 0.

Napisz w Javie program działający w czasie $O((\max(n, m))^3)$, który oblicza maksymalną wartość $msum(i, j, k, l)$ oraz wyznacza maksymalną podtablicę $mstab = a[i..j][k..l]$ o najmniejszej liczbie elementów, której indeksy: i, j, k, l tworzą ciąg leksykograficznie najmniejszy.

Wejście

Dane do programu wczytywane są ze standardowego wejścia (klawiatury) zgodnie z poniższą specyfikacją. Pierwszą podawaną wartością jest dodatnia liczba całkowita oznaczająca ilość zestawów danych, po której na wejściu pojawią się zestawy danych w ilości równej wczytanej liczbie. Każdy zestaw danych zawiera w pierwszej linii numer zestawu (z) od 1 do 100, po którym występuje ciąg znaków " : " oraz dwie dodatnie liczby całkowite z zakresu od 1 do 100, oznaczające odpowiednio liczbę wierszy oraz liczbę kolumn tablicy, w następnych liniach podawane są dane będące kolejnymi wierszami tablicy zgodnie z podaną liczbą wierszy i kolumn. Dane każdego zestawu są liczbami całkowitymi z zakresu od -2^{15} do $+2^{15}$.

Ostatnia linia każdego zestawu zakończona jest znakiem '\n'.

Wyjście

Dla każdego numeru zestawu danych jeśli maksymalna podtablica nie jest pusta, program wypisze linie postaci:

```
z: mstab = a[i..j][k..l], msum = msum(i, j, k, l)
    przy czym: i, j, k, l, wyznaczają maksymalną podtablicę a[i..j][k..l]
```


w przeciwnym przypadku program powinien wypisywać tekst:

```
z: mstab is empty
```

Wymagania implementacyjne

W pierwszej linii program powinien zawierać komentarz:

1. // Imie Nazwisko – nr grupy a
2. Jedyńm dozwolonym importem jest obsługa wczytywania z klawiatury, to jest:
import java.util.Scanner;

	<p style="text-align: center;">Metody programowania 2023/2024</p> <p style="text-align: center;">Maksymalna podtablica 2D</p> <p style="text-align: center;">o najmniejszej liczbie elementów</p>	<p style="text-align: center;">P_01</p>
---	--	---

3. Główna klasa musi nazywać się *Source*, co oznacza ogólne ramy kodu postaci:

```
class Source {
    public static void main( String [] args ) {
        ...
    }
}
```

4. Wczytywanie musi się odbywać przez pojedynczą zmienną klasy *Scanner*, zadeklarowaną zewnętrznie w stosunku do wszystkich metod głównej klasy.

W praktyce oznacza to tylko jedną deklarację w przykładowej postaci, np.:

public static Scanner scanner = new Scanner(System.in); w pierwszej linii ciała głównej klasy.

5. Kod programu powinien zawierać komentarze i testy zgodnie z opisem podanym w Regulaminie zaliczania programów na BaCy.

Dane przykładowe

<u>Wejście:</u>	<u>Wyjście:</u>
<pre>6 1 : 1 6 -2 7 -4 8 -5 4 2 : 2 5 1 1 -1 -1 0 1 1 -1 -1 4 3 : 2 5 0 -1 -1 1 1 4 -2 -2 1 1 4 : 2 5 0 -1 -1 4 0 4 -2 -2 0 0 5 : 2 5 -1 -2 -3 -1 -2 -1 -1 -1 -1 -5 6 : 2 5 0 0 0 0 0 0 0 0 0 0</pre>	<pre>1: ms_tab = a[0..0][1..5], msum=39 2: ms_tab = a[0..1][0..4], msum=16 3: ms_tab = a[1..1][0..0], msum=12 4: ms_tab = a[0..0][3..3], msum=12 5: ms_tab is empty 6: ms_tab = a[0..0][0..0], msum=0</pre>